

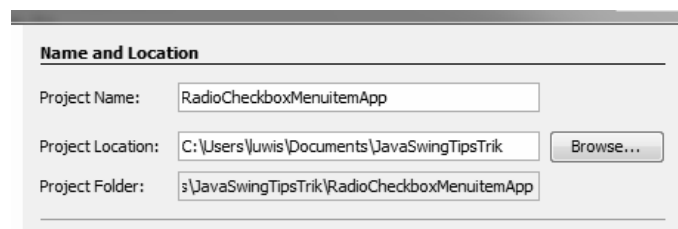


Menu Item dengan RadioButton dan CheckBox

Pada bab ini, kita akan membuat menu dengan menu itemnya berupa radio button dan check box. Java sudah menyediakan menu item seperti ini, yaitu `JRadioButtonMenuItem` dan `JCheckBoxMenuItem`. Penggunaan kedua jenis menu item ini sama seperti `JMenuItem` biasa, namun sedikit berbeda untuk `JRadioButtonMenuItem`.

Berikut langkah-langkahnya:

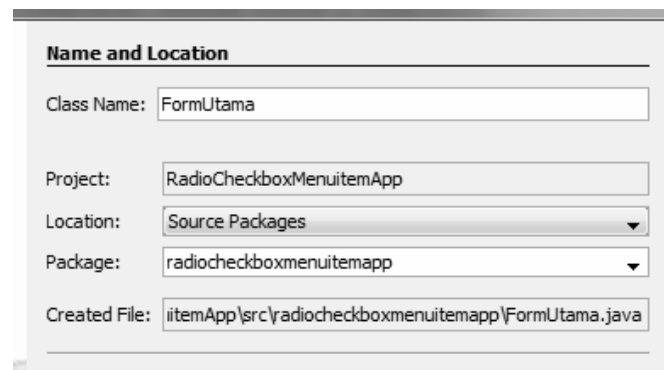
1. Buka Netbeans IDE.
2. Buat project baru. Klik menu File > New Project.
3. Pilih Categories: Java dan Project: Java Application. Kemudian klik Next.
4. Beri nama project **RadioCheckboxMenuItemApp**. Kemudian klik Finish.



Gambar 7.1 Membuat Project RadioCheckboxMenuItemApp

5. Kita buat sebuah Form.

6. Klik kanan project RadioCheckboxMenuItemApp > New > Java Class.
7. Beri nama class **FormUtama** dan package **radiocheckboxmenuItemapp**. Kemudian klik Finish.



Gambar 7.2 Membuat Class FormUtama

8. Ketik kode berikut pada class FormUtama.

```
package radiocheckboxmenuItemapp;

import java.awt.Dimension;
import javax.swing.ButtonGroup;
import javax.swing.JCheckBoxMenuItem;
import javax.swing.JFrame;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JRadioButtonMenuItem;

public class FormUtama extends JFrame {

    JMenuBar menuBar;

    public FormUtama(){
        super("RadioButton dan CheckBox Menu Item");
        initComponents();
    }

    public void initComponents() {
        menuBar = new JMenuBar();
        setCheckBoxMenuItem();
        setRadioButtonMenuItem();
        setJMenuBar(menuBar);
        setPreferredSize(new Dimension(500, 200));
        pack();
        setVisible(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

```

    }

    public void setCheckBoxMenuItem() {
        JMenu menuStyle;
        JCheckBoxMenuItem itemBold, itemItalic;

        menuStyle = new JMenu("Style");

        itemItalic = new JCheckBoxMenuItem("Italic");
        menuStyle.add(itemItalic);
        itemBold = new JCheckBoxMenuItem("Bold");
        menuStyle.add(itemBold);

        menuBar.add(menuStyle);
    }

    public void setRadioButtonMenuItem() {
        JMenu menuColor;
        JRadioButtonMenuItem itemRed, itemBlue, itemGreen;
        ButtonGroup groupColor;

        menuColor = new JMenu("Color");
        groupColor = new ButtonGroup();

        itemRed = new JRadioButtonMenuItem("Red");
        groupColor.add(itemRed);
        menuColor.add(itemRed);

        itemBlue = new JRadioButtonMenuItem("Blue");
        groupColor.add(itemBlue);
        menuColor.add(itemBlue);

        itemGreen = new JRadioButtonMenuItem("Green");
        groupColor.add(itemGreen);
        menuColor.add(itemGreen);

        menuBar.add(menuColor);
    }
}

```

Penjelasan Program

Cara menggunakan JCheckBoxMenuItem ini sama seperti JMenuItem biasa. Lihat method setCheckBoxMenuItem(), JCheckBoxMenuItem hanya tinggal ditambahkan pada JMenu saja (menuStyle).

Cara menggunakan JRadioButtonMenuItem sedikit berbeda. Komponen radio button merupakan komponen input pemilihan di mana kita hanya bisa memilih satu item dalam group yang sama.

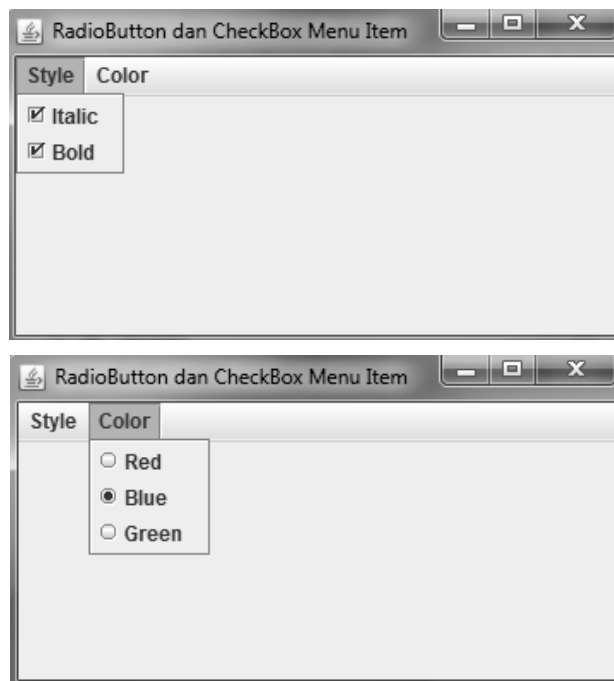
Lihat method `setRadioButtonMenuItem()`. Semua `JRadioButtonMenuItem` ditambahkan dalam satu group. Sehingga jika kita telah memilih satu menu item, maka menu item lain tidak akan ikut terpilih.

9. Buka class **Main.java**. Ketik kode berikut.

```
package radiocheckboxmenumitemapp;  
  
public class Main {  
    public static void main(String[] args) {  
        new FormUtama();  
    }  
}
```

10. Jalankan project `RadioCheckboxMenuItemApp`.

11. Klik kanan project `RadioCheckboxMenuItemApp` > Run.



Gambar 7.3 Aplikasi RadioCheckboxMenuItemApp

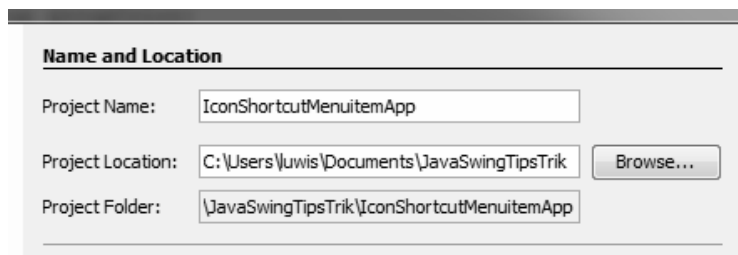


Membuat Ikon dan Shortcut pada JMenuitem

Untuk memberikan suatu tampilan berbeda pada JMenuitem, kita bisa menambahkan sebuah ikon dalam JMenuitem. Kita juga dapat membuat shortcut, dengan tujuan mempercepat pemanggilan fungsi pada JMenuitem. Pada bab ini, kita akan membahas bagaimana menambah ikon dan membuat shortcut pada JMenuitem.

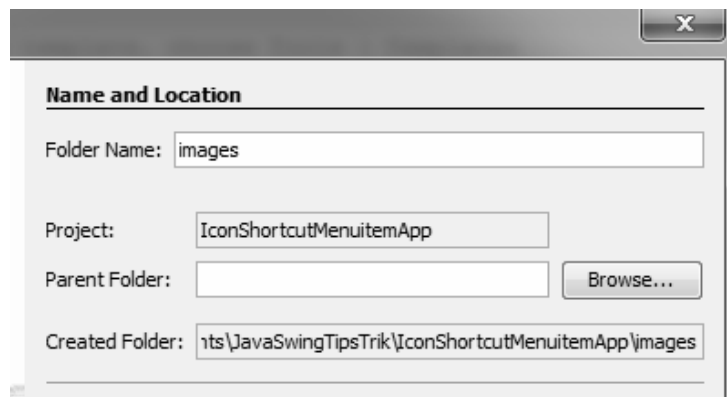
Berikut langkah-langkahnya:

1. Buka Netbeans IDE.
2. Buat project baru. Klik menu File > New Project.
3. Pilih Categories: Java dan Project: Java Application. Kemudian klik Next.
4. Beri nama project **IconShortcutMenuitemApp**. Kemudian klik Finish.



Gambar 8.1 Membuat Project IconShortcutMenuitemApp

5. Kita buat folder baru untuk meletakkan image ikon kita.
6. Klik kanan project IconShortcutMenuItemApp > New > Other.
7. Pilih Categories: Other dan File Types: Folder. Kemudian klik Next.
8. Beri nama folder **images**. Kemudian klik Finish.



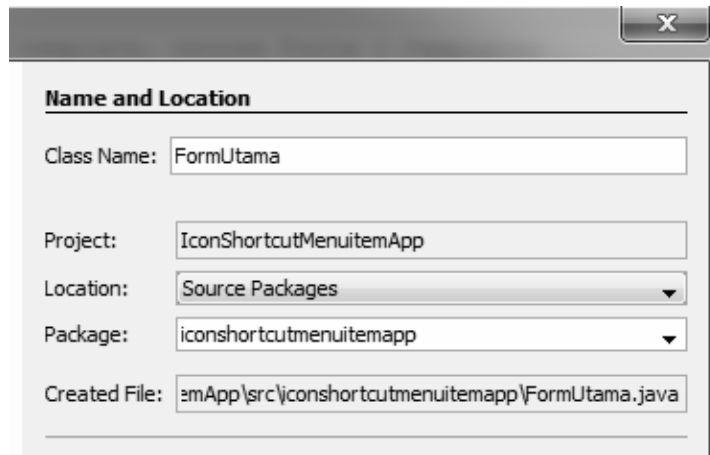
Gambar 8.2 Membuat Folder images

9. Copy image **close.png** dalam Bonus CD buku ini ke dalam folder images pada project.



Gambar 8.3 Image close.png

10. Kita buat sebuah Form.
11. Klik kanan project IconShortcutMenuItemApp > New > Java Class.
12. Beri nama class **FormUtama** dan package **iconshortcutmenuitemapp**. Kemudian klik Finish.



Gambar 8.4 Membuat Class FormUtama

13. Ketik kode berikut pada class FormUtama.

```
package iconshortcutmenuitemapp;

import java.awt.Dimension;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.InputEvent;
import java.awt.event.KeyEvent;
import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.KeyStroke;

public class FormUtama extends JFrame {

    JMenuItem itemQuit;
    JMenu menu;
    JMenuBar menuBar;

    public FormUtama(){
        super("Icon dan Shortcut pada Menu Item");
        initComponents();
    }

    public void initComponents() {
        menuBar = new JMenuBar();
        menu = new JMenu("File");

        setIconAndShortcutMenuItem();

        menu.add(itemQuit);
    }
}
```

```

        menuBar.add(menu);
        setJMenuBar(menuBar);
        setPreferredSize(new Dimension(500, 200));
        pack();
        setVisible(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

    public void setIconAndShortcutMenuItem(){
        itemQuit = new JMenuItem("Close Window");
        itemQuit.setIcon(new ImageIcon("images/close.png"));

        itemQuit.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_Q,
            InputEvent.CTRL_MASK));
        itemQuit.addActionListener(new ActionListener() {

            public void actionPerformed(ActionEvent ae) {
                System.exit(0);
            }
        });
    }
}

```

Penjelasan Program

Kita hanya membuat sebuah JMenuItem, yaitu itemQuit. JMenuItem ini kita beri ikon “close.png”.

Lihat method setAccelerator(). Kita beri juga shortcut berupa kombinasi “CTRL + Q” untuk memanggil action dari JMenuItem itemQuit untuk menutup Window.

14. Buka class Main.java. Ketik kode berikut.

```

package iconshortcutmenumitemapp;

public class Main {

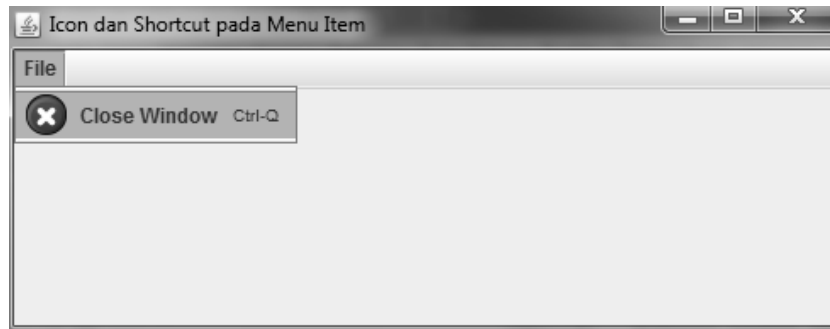
    public static void main(String[] args) {
        new FormUtama();
    }
}

```

15. Jalankan project IconShortcutMenuitemApp.

16. Klik kanan project IconShortcutMenuitemApp > Run.

17. Klik menu item **Close Window** atau tekan shortcut **CTRL+Q**.



Gambar 8.5 Aplikasi IconShortcutMenuItemApp

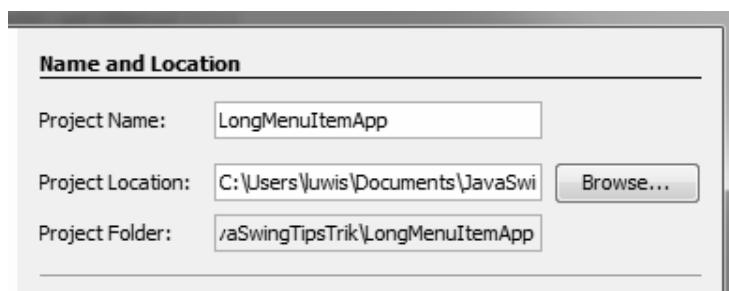


Mengatasi Menu Item yang Banyak

Sebuah menu dengan banyak menu item, mungkin tidak menjadi masalah, karena JMenu tidak membatasi jumlah maksimum JMenuitem yang dapat ditambahkan. Namun, bagi user bisa jadi kurang nyaman melihat popup menu yang sangat panjang. Untuk menyederhanakan tampilan popup menu, kita bisa mengatur layout popup sendiri, sehingga dapat menampung banyak menu item tapi tetap terlihat rapi.

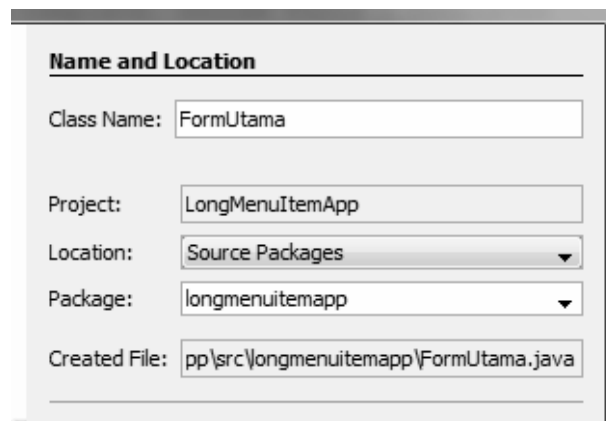
Berikut langkah-langkahnya:

1. Buka Netbeans IDE.
2. Buat project baru. Klik menu File > New Project.
3. Pilih Categories: Java dan Project: Java Application. Kemudian klik Next.
4. Beri nama project **LongMenuItemApp**. Kemudian klik Finish.



Gambar 9.1 Membuat Project LongMenuItemApp

5. Kita buat sebuah Form.
6. Klik kanan project LongMenuItemApp > New > Java Class.
7. Beri nama class **FormUtama** dan package **longmenuitemapp**.
Kemudian klik Finish.



Gambar 9.2 Membuat Class FormUtama

8. Ketik kode berikut pada class **FormUtama**.

```
package longmenuitemapp;

import java.awt.Dimension;
import java.awt.GridLayout;
import javax.swing.JFrame;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;

public class FormUtama extends JFrame {

    JMenu menu;
    JMenuBar menuBar;

    public FormUtama() {
        super("long Item Menu");
        initComponents();
    }

    public void initComponents() {
        setMenu();
        setPreferredSize(new Dimension(400, 400));
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        pack();
        setVisible(true);
    }
}
```

```

    }

    public void setMenu() {
        menu = new JMenu("Menu");
        menu.getPopupMenu().setLayout(new GridLayout(0, 4));
        for (int i = 0; i < 30; i++) {
            menu.add(new JMenuItem("item " + (i + 1)));
        }
        menuBar = new JMenuBar();
        menuBar.add(menu);
        setJMenuBar(menuBar);
    }
}

```

Penjelasan Program

Lihat pada method `setMenu()`. Kita ubah layout popup menu menjadi `GridLayout` dengan 5 kolom. Selanjutnya kita buat 30 `JMenuItem` ke dalam `JMenu` tersebut.

9. Buka class **Main.java**. Ketik kode berikut.

```

package longmenuitemapp;

public class Main {

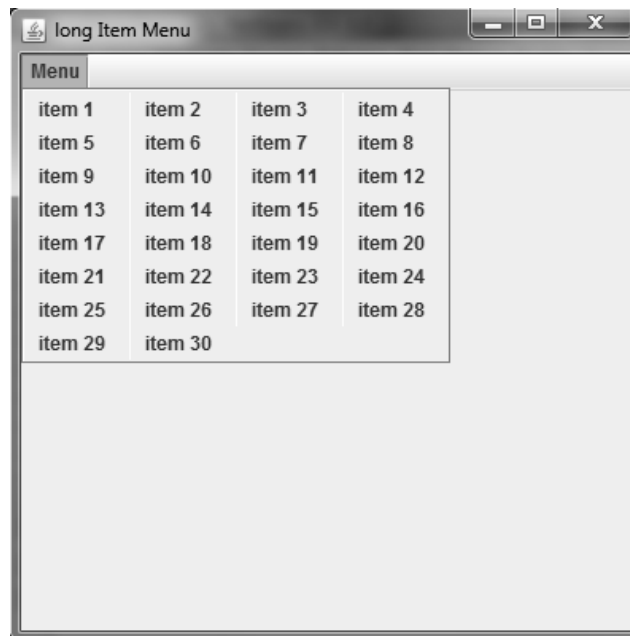
    public static void main(String[] args) {
        new FormUtama();
    }
}

```

10. Jalankan project `LongMenuItemApp`.

11. Klik kanan project `LongMenuItemApp` > Run.

12. Klik menu **Menu**.



Gambar 9.3 Aplikasi LongMenuItemApp

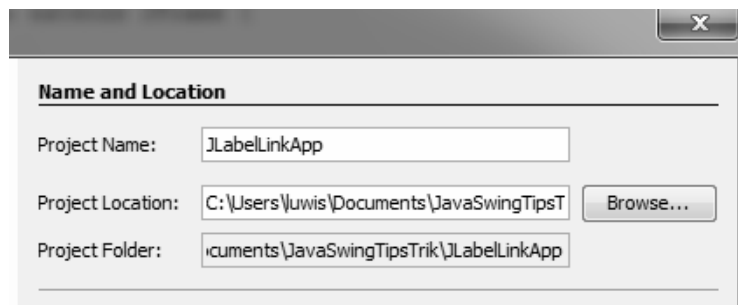
10

Hyperlink pada JLabel

Secara sederhana fungsi JLabel hanyalah menampilkan label/teks. Pada bab ini kita akan mengubah JLabel menjadi hyperlink. Caranya kita tambahkan mouse listener, sehingga jika JLabel diklik maka akan memicu aplikasi untuk membuka web browser menuju suatu halaman web tertentu.

Berikut langkah-langkahnya:

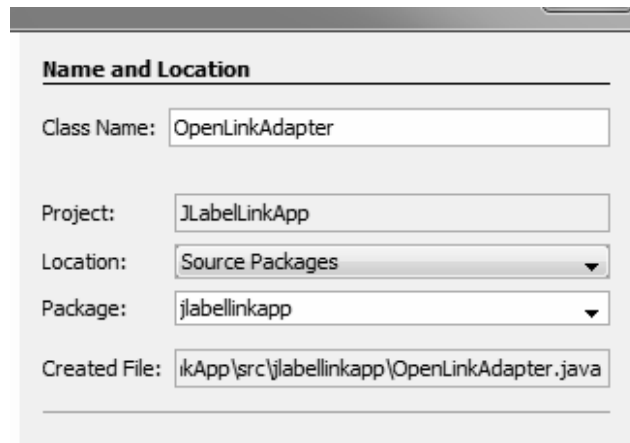
1. Buka Netbeans IDE.
2. Buat project baru. Klik menu File > New Project.
3. Pilih Categories: Java dan Project: Java Application. Kemudian klik Next.
4. Beri nama project **JLabelLinkApp**. Kemudian klik Finish.



Gambar 10.1 Membuat Project JLabelLinkApp

5. Kita buat sebuah mouse adapter sebagai action ketika JLabel diklik.

6. Klik kanan project JLabelLinkApp > New > Java Class.
7. Beri nama class **OpenLinkAdapter** dan package **jlabellinkapp**. Kemudian klik Finish.



Gambar 10.2 Membuat Class OpenLinkAdapter

8. Ketik kode berikut pada class **OpenLinkAdapter**.

```
package jlabellinkapp;

import java.awt.Desktop;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.net.URI;

public class OpenLinkAdapter extends MouseAdapter {

    String url;

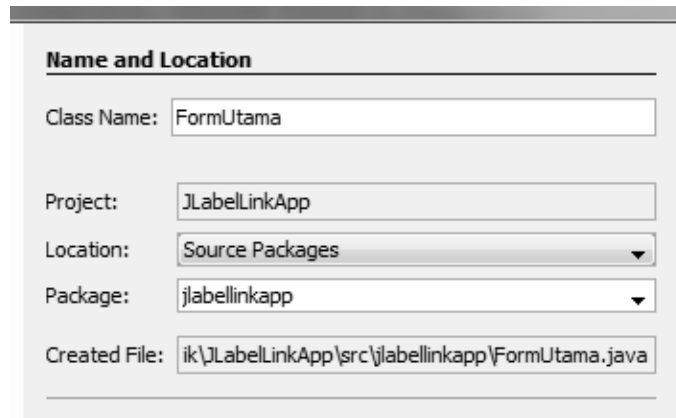
    public OpenLinkAdapter(String url) {
        this.url = url;
    }

    @Override
    public void mouseClicked(MouseEvent evt) {
        try {
            Desktop desktop = java.awt.Desktop.getDesktop();
            URI uri = new java.net.URI(url);
            desktop.browse(uri);
        } catch (Exception e) {
        }
    }
}
```

Penjelasan Program

Lihat pada method mouseClicked(). Adapter akan menjalankan default web browser yang ada di komputer kita untuk membuka halaman web sesuai dengan URL yang ingin kita tuju.

9. Kita buat sebuah Form.
10. Klik kanan project JLabelLinkApp > New > Java Class.
11. Beri nama class **FormUtama** dan package **jlabellinkapp**. Kemudian klik Finish.



Gambar 10.3 Membuat Class FormUtama

12. Ketik kode berikut pada class **FormUtama**.

```
package jlabellinkapp;

import java.awt.Color;
import java.awt.Cursor;
import java.awt.Dimension;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.SwingConstants;

public class FormUtama extends JFrame {

    JLabel label;

    public FormUtama() {
        super("JLabel Hyperlink");
        initComponents();
    }
}
```



```

public void initComponents() {
    setLabelHyperlink();
    setPreferredSize(new Dimension(300, 100));
    pack();
    setVisible(true);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}

public void setLabelHyperlink() {
    label = new JLabel();
    label.setText("<HTML><U>Klik untuk menuju  
GOOGLE.COM</U></HTML>");
    label.setForeground(Color.BLUE);
    label.setHorizontalAlignment(SwingConstants.CENTER);
    label.setCursor(new Cursor(Cursor.HAND_CURSOR));
    label.addMouseListener(new
OpenLinkAdapter("http://www.google.com"));
    add(label);
}
}

```

Penjelasan Program

Lihat pada method `setLabelHyperlink()`. Teks pada `JLabel` kita tambahkan dengan syntax HTML untuk membuat garis underline pada teks. Pada `JLabel`, kita beri action agar jika diklik akan membuka halaman web `google.com`.

13. Buka class **Main.java**. Ketik kode berikut.

```

package jlabellinkapp;

public class Main {

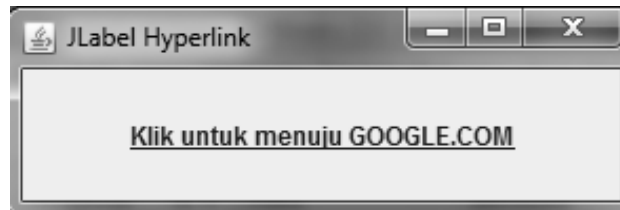
    public static void main(String[] args) {
        new FormUtama();
    }
}

```

14. Jalankan project `JLabelLinkApp`.

15. Klik kanan project `JLabelLinkApp` > Run.

16. Klik pada **JLabel**. Web browser akan membuka menuju halaman `GOOGLE.COM`.



Gambar 10.4 Aplikasi JLabelLinkApp



Gambar 10.5 Halaman Web Google.com



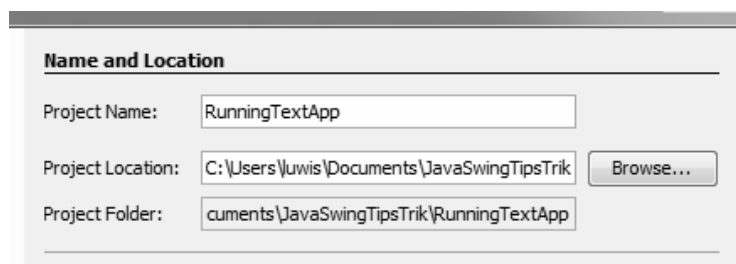
Running Teks

Jika kita melihat pada web portal berita, sering kita melihat ada running teks di mana sebuah teks berjalan dari satu sisi ke sisi yang lain. Pada bab ini, kita akan coba mengaplikasikan running teks pada Java.

Di Java, JLabel tidak memiliki method untuk membuat running teks seperti itu, tapi hal ini bisa kita lakukan dengan cara menggeser karakter-karakter teks pada JLabel secara periodik namun cepat, sehingga teks seolah-olah sedang berjalan.

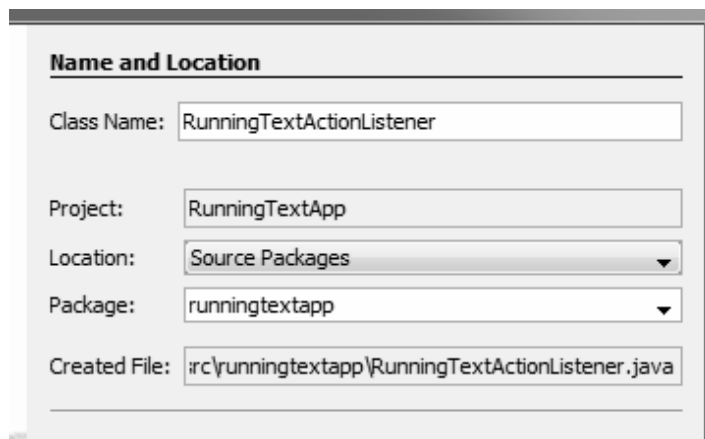
Berikut langkah-langkahnya:

1. Buka Netbeans IDE.
2. Buat project baru. Klik menu File > New Project.
3. Pilih Categories: Java dan Project: Java Application. Kemudian klik Next.
4. Beri nama project **RunningTextApp**. Kemudian klik Finish.



Gambar 11.1 Membuat Project RunningTextApp

13. Kita buat listener untuk menggerakkan teks kita.
14. Klik kanan project RunningTextApp > New > Java Class.
15. Beri nama class **RunningTextActionListener** dan package **runningtextapp**. Kemudian klik Finish.



Gambar 11.2 Membuat Class RunningTextApp

16. Ketik kode berikut pada class **RunningTextActionListener**.

```
package runningtextapp;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JLabel;

public class RunningTextActionListener implements ActionListener
{
    JLabel label;

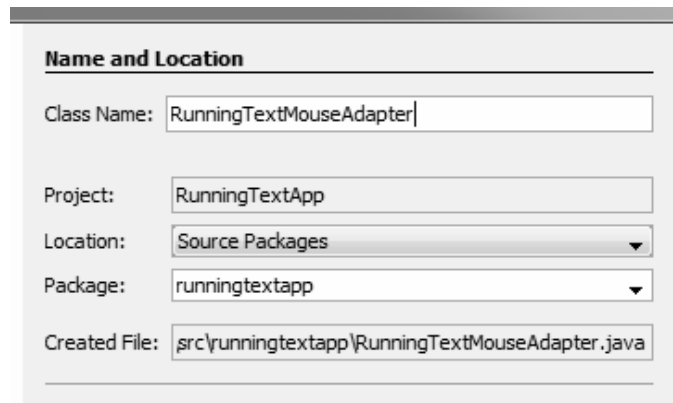
    public RunningTextActionListener(JLabel label) {
        this.label = label;
    }

    public void actionPerformed(ActionEvent e) {
        String oldText = label.getText();
        String newText = oldText.substring(1) +
oldText.substring(0, 1);
        label.setText(newText);
    }
}
```

Penjelasan Program

RunningTextActionListener adalah listener yang akan menggerakkan teks kita. Teks pada JLabel digeser dengan cara karakter terdepan dipindahkan ke akhir teks. Dengan cara ini, teks seolah berjalan dari kanan ke kiri.

17. Kita buat mouse listener agar saat mouse berada di atas JLabel, maka running teks akan berhenti berjalan.
18. Klik kanan project RunningTextApp > New > Java Class.
19. Beri nama class **RunningTextMouseAdapter** dan package **runningtextapp**. Kemudian klik Finish.



Gambar 11.3 Membuat Class RunningTextMouseAdapter

20. Ketik kode berikut pada class RunningTextMouseAdapter.

```
package runningtextapp;

import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import javax.swing.Timer;

public class RunningTextMouseAdapter extends MouseAdapter {

    Timer timer;

    public RunningTextMouseAdapter(Timer timer) {
        this.timer = timer;
    }

    public void mouseEntered(MouseEvent e) {
        timer.stop();
    }
}
```

```

    }

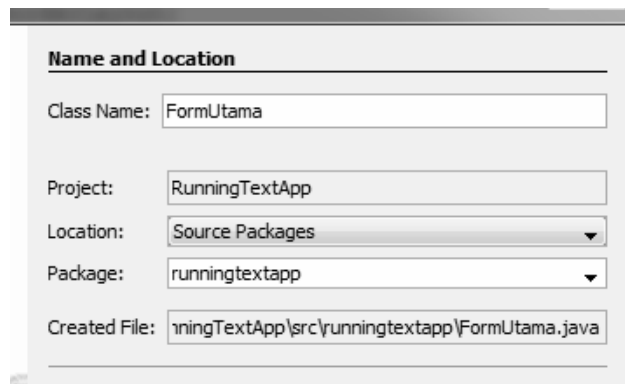
    public void mouseExited(MouseEvent e) {
        timer.start();
    }
}

```

Penjelasan Program

RunningTextMouseAdapter adalah MouseListener yang akan mendeteksi posisi Mouse. Listener RunningTextActionListener yang kita buat sebelumnya akan diaktifkan dengan timer. Ketika mouse berada di atas JLabel (mouseEntered()) maka timer akan berhenti dan ketika mouse tidak berada di atas JLabel (mouseExited()) maka timer akan berjalan kembali.

21. Kita buat sebuah Form.
22. Klik kanan project RunningTextApp > New > Java Class.
23. Beri nama class **FormUtama** dan package **runningtextapp**. Kemudian klik Finish.



Gambar 11.4 Membuat Class FormUtama

24. Ketik kode berikut pada class **FormUtama**.

```

package runningtextapp;

import java.awt.BorderLayout;
import javax.swing.Timer;
import javax.swing.JFrame;

```

```

import javax.swing.JLabel;

public class FormUtama extends JFrame {

    JLabel label;
    Timer timer;

    public FormUtama(){
        super("Running Text");
        initComponents();
        enableRunningText();
    }

    public void initComponents() {
        label = new JLabel("Spanyol juara EURO 2012 setelah
mengalahkan Italia di final ");
        getContentPane().add(label, BorderLayout.CENTER);
        pack();
        setVisible(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

    public void enableRunningText() {
        timer = new Timer(100, new
RunningTextActionListener(label));
        label.addMouseListener(new
RunningTextMouseAdapter(timer));
        timer.start();
    }
}

```

Penjelasan Program

Lihat pada method `enableRunningText()`. `RunningTextActionListener` yang akan menggerakkan teks, kita aktifkan dengan `Timer`. `Timer` ini kita atur nilainya sebesar 100, artinya teks pada `JLabel` akan bergerak setiap 100 ms atau 0,1 detik.

25. Buka class **Main.java**. Ketik kode berikut.

```

package runningtextapp;

public class Main {

    public static void main(String[] args) {
        new FormUtama();
    }
}

```

26. Jalankan project RunningTextApp.
27. Klik kanan project RunningTextApp > Run.
28. Sekarang teks seperti sedang berjalan.



Gambar 11.5 Aplikasi RunningTextApp

12₃

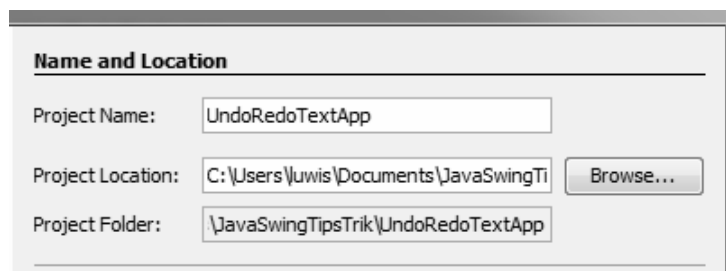
Undo/Redo pada Teks Input

Di Java, kita tidak bisa melakukan Undo atau Redo pada komponen teks input seperti `TextField` dan `TextArea`. Padahal penggunaan Undo/Redo sebenarnya sangat umum digunakan, misalnya untuk melakukan pembenaran kesalahan ketik.

Meski tidak mendukung fungsi Undo/Redo, namun kita bisa membuat sendiri kedua fungsi ini dengan memanfaatkan class `UndoManager` yang akan kita implementasikan pada `TextField` atau `TextArea`.

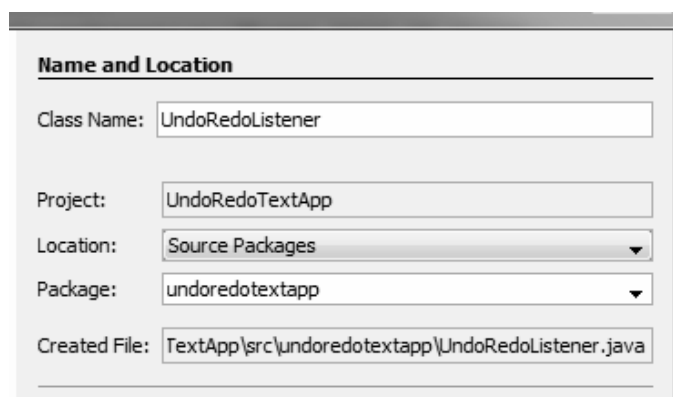
Berikut langkah-langkahnya:

1. Buka Netbeans IDE.
2. Buat project baru. Klik menu File > New Project.
3. Pilih Categories: Java dan Project: Java Application. Kemudian klik Next.
4. Beri nama project **UndoRedoTextApp**. Kemudian klik Finish.



Gambar 12.1 Membuat Project UndoRedoTextApp

5. Kita buat listener untuk mendeteksi aktivitas apa yang kita lakukan dalam teks input (JTextField/JTextArea), misal kita mengetik, menghapus huruf, atau melakukan copy/paste teks dalam teks input.
6. Klik kanan project UndoRedoTextApp > New > Java Class.
7. Beri nama class **UndoRedoListener** dan package **undoredotextapp**. Kemudian klik Finish.



Gambar 12.2 Membuat Class *UndoRedoListener*

8. Ketik kode berikut pada class **UndoRedoListener**.

```
package undoredotextapp;

import javax.swing.event.UndoableEditEvent;
import javax.swing.event.UndoableEditListener;
import javax.swing.undo.UndoManager;

public class UndoRedoListener implements UndoableEditListener {
    UndoManager undoRedoManager;

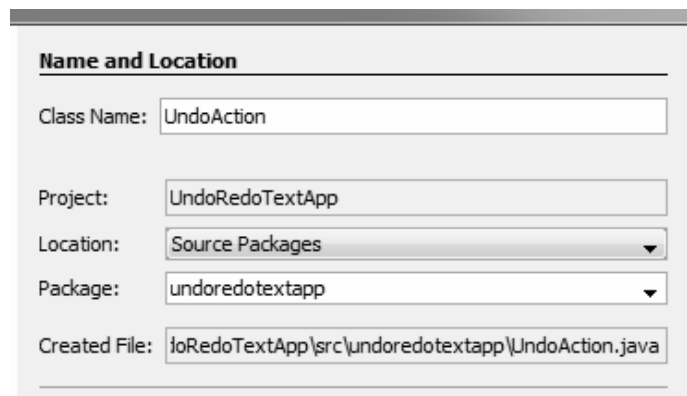
    public UndoRedoListener(UndoManager undoRedoManager) {
        this.undoRedoManager = undoRedoManager;
    }

    public void undoableEditHappened(UndoableEditEvent evt) {
        undoRedoManager.addEdit(evt.getEdit());
    }
}
```

Penjelasan Program

Setiap aktivitas dalam teks input akan menghasilkan perubahan dalam teks input itu. UndoRedoListener mendeteksi perubahan di dalam teks input yang kemudian akan disimpan oleh UndoManager. Sehingga saat kita melakukan Undo/Redo, maka teks input dapat kembali ke kondisi yang kita maksud.

9. Kita buat masing-masing sebuah action untuk melakukan Undo dan Redo.
10. Klik kanan project UndoRedoTextApp > New > Java Class.
11. Beri nama class **UndoAction** dan package **undoredotextapp**. Kemudian klik Finish.



Gambar 12.3 Membuat Class UndoAction

12. Ketik kode berikut pada class **UndoAction**.

```
package undoredotextapp;

import java.awt.event.ActionEvent;
import javax.swing.AbstractAction;
import javax.swing.undo.CannotUndoException;
import javax.swing.undo.UndoManager;

public class UndoAction extends AbstractAction {

    UndoManager undoRedoManager;

    public UndoAction(UndoManager undoRedoManager) {
        this.undoRedoManager = undoRedoManager;
    }
}
```

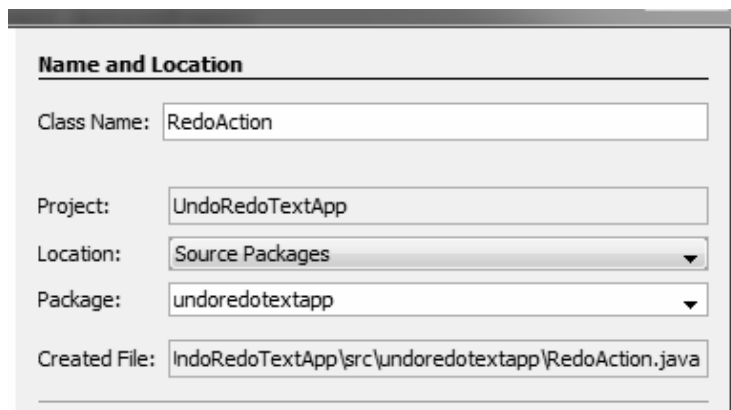
```

    public void actionPerformed(ActionEvent evt) {
        try {
            if (undoRedoManager.canUndo()) {
                undoRedoManager.undo();
            }
        } catch (CannotUndoException e) {
        }
    }
}

```

13. Klik kanan project `UndoRedoTextApp` > New > Java Class.

14. Beri nama class **RedoAction** dan package `undoredotextapp`.
Kemudian klik Finish.



Gambar 12.4 Membuat Class RedoAction

15. Ketik kode berikut pada class **RedoAction**.

```

package undoredotextapp;

import java.awt.event.ActionEvent;
import javax.swing.AbstractAction;
import javax.swing.undo.CannotUndoException;
import javax.swing.undo.UndoManager;

public class RedoAction extends AbstractAction {

    UndoManager undoRedoManager;

    public RedoAction(UndoManager undoRedoManager) {
        this.undoRedoManager = undoRedoManager;
    }
}

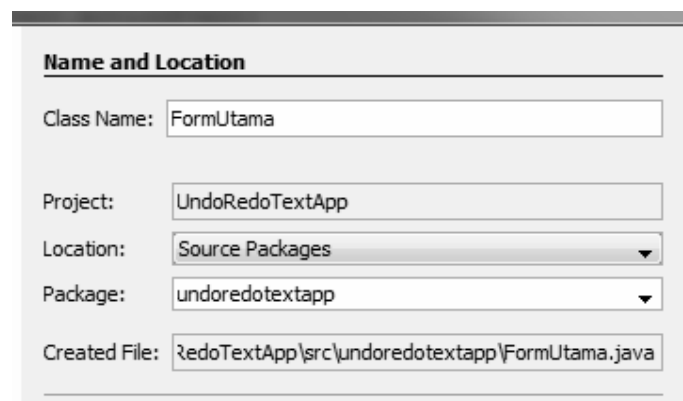
```

```

        public void actionPerformed(ActionEvent evt) {
            try {
                if (undoRedoManager.canRedo()) {
                    undoRedoManager.Redo();
                }
            } catch (CannotUndoException e) {
            }
        }
    }
}

```

16. Kita buat sebuah Form.
17. Klik kanan project UndoRedoTextApp > New > Java Class.
18. Beri nama class **FormUtama** dan package **undoredotextapp**. Kemudian klik Finish.



Gambar 12.5 Membuat Class FormUtama

19. Ketik kode berikut pada class FormUtama.

```

package undoredotextapp;

import java.awt.BorderLayout;
import java.awt.Dimension;
import java.awt.Toolkit;
import java.awt.event.KeyEvent;
import javax.swing.JFrame;
import javax.swing.JTextArea;
import javax.swing.KeyStroke;
import javax.swing.undo.UndoManager;

public class FormUtama extends JFrame {

    JTextArea textArea;

```

```

final UndoManager undoRedoManager = new UndoManager();

public FormUtama() {
    super("Undo/Redo Text");
    initComponents();
}

public void initComponents() {
    textArea = new JTextArea();
    textArea.getDocument().addUndoableEditListener(new
UndoRedoListener(undoRedoManager));
    textArea.setPreferredSize(new Dimension(300, 200));
    setUndo();
    setRedo();
    getContentPane().add(textArea, BorderLayout.CENTER);

    pack();
    setVisible(true);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}

public void setUndo() {
    textArea.getActionMap().put("Undo", new
UndoAction(undoRedoManager));

    textArea.getInputMap().put(KeyStroke.getKeyStroke(KeyEvent.VK_Z,
Toolkit.getDefaultToolkit().getMenuShortcutKeyMask()), "Undo");
}

public void setRedo() {
    textArea.getActionMap().put("Redo", new
RedoAction(undoRedoManager));

    textArea.getInputMap().put(KeyStroke.getKeyStroke(KeyEvent.VK_Y,
Toolkit.getDefaultToolkit().getMenuShortcutKeyMask()), "Redo");
}
}

```

Penjelasan Program

Lihat method `setUndo()`. Action undo kita petakan dengan identifier "Undo". Kita atur juga supaya ketika kita menekan shortcut CTRL+Z, maka action undo dapat dijalankan.

Lihat method `setRedo()`. Action redo kita petakan dengan identifier "Redo". Kita atur juga supaya ketika kita menekan shortcut CTRL+Y, maka action redo dapat dijalankan.

20. Buka class **Main.java**. Ketik kode berikut.

```

package undoredotextapp;

public class Main {

```

```
public static void main(String[] args) {  
    new FormUtama();  
}
```

21. Jalankan project UndoRedoTextApp.
22. Klik kanan project UndoRedoTextApp > Run.
23. Coba ketik beberapa karakter dan tekan CTRL+Z atau CTRL+Y untuk melakukan undo/redo.



Gambar 12.6 Aplikasi UndoRedoTextApp

13

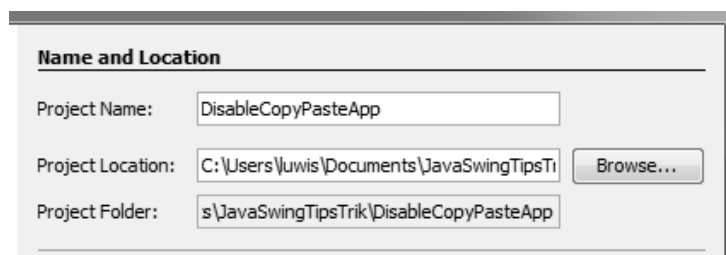
Menonaktifkan Copy/Paste pada Teks Input

Untuk suatu alasan, adakalanya kita ingin agar Form aplikasi kita tidak dapat melakukan aktivitas Copy dan Paste. Jika pada bab sebelumnya kita telah membuat fungsi Redo/Undo, sekarang kita justru akan menghilangkan fungsi Copy/Paste pada JTextField dan JTextArea.

Fungsi Copy/Paste merupakan fungsi asli yang memang sudah terdefinisi pada JTextField dan JTextArea. Cara termudah untuk menonaktifkannya, yaitu dengan mendefinisikan ulang kedua fungsi ini.

Berikut langkah-langkahnya:

1. Buka Netbeans IDE.
2. Buat project baru. Klik menu File > New Project.
3. Pilih Categories: Java dan Project: Java Application. Kemudian klik Next.
4. Beri nama project **DisableCopyPasteApp**. Kemudian klik Finish.



Gambar 13.1 Membuat Project DisableCopyPasteApp

5. Kita akan membuat sebuah TextArea baru yang merupakan turunan dari class JTextArea. Namun TextArea yang baru ini tidak bisa melakukan aktivitas copy paste.
6. Klik kanan project DisableCopyPasteApp > New > Java Class.
7. Beri nama class **NoCopasTextArea** dan package **disablecopypasteapp**. Kemudian klik Finish.



Gambar 13.2 Membuat Class NoCopasTextArea

8. Ketik kode berikut pada class **NoCopasTextArea**.

```
package disablecopypasteapp;
import javax.swing.JTextArea;

public class NoCopasTextArea extends JTextArea {

    @Override
    public void paste() {
    }

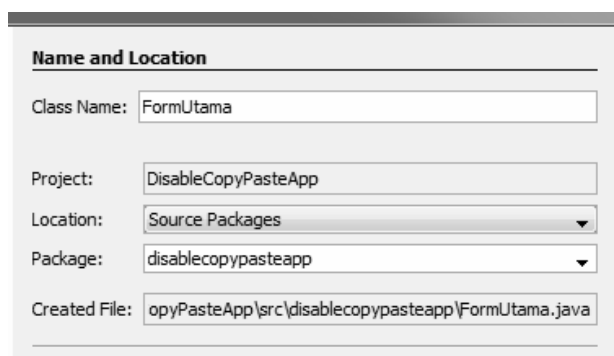
    @Override
    public void copy() {
    }
}
```

Penjelasan Program

Pada NoCopasTextArea kita override method asli copy() dan paste() dari JTextArea. Di dalam kedua method copy() dan paste() tersebut, kita biarkan kosong begitu saja. Tujuannya agar

ketika kita melakukan copy (CTRL+C) atau paste (CTRL+V) pada TextArea ini, maka kedua fungsi itu tidak akan melakukan action apa pun.

9. Kita buat sebuah Form.
10. Klik kanan project DisableCopyPasteApp > New > Java Class.
11. Beri nama class **FormUtama** dan package **disablecopypasteapp**. Kemudian klik Finish.



Gambar 13.3 Membuat Class FormUtama

12. Ketik kode berikut pada class **FormUtama**.

```
package disablecopypasteapp;

import java.awt.BorderLayout;
import javax.swing.JFrame;

public class FormUtama extends JFrame{

    NoCopasTextArea textArea;

    public FormUtama() {
        super("Disable Copy Paste");
        initComponents();
    }

    public void initComponents() {
        textArea = new NoCopasTextArea();
        getContentPane().add(textArea, BorderLayout.CENTER);
        pack();
        setVisible(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

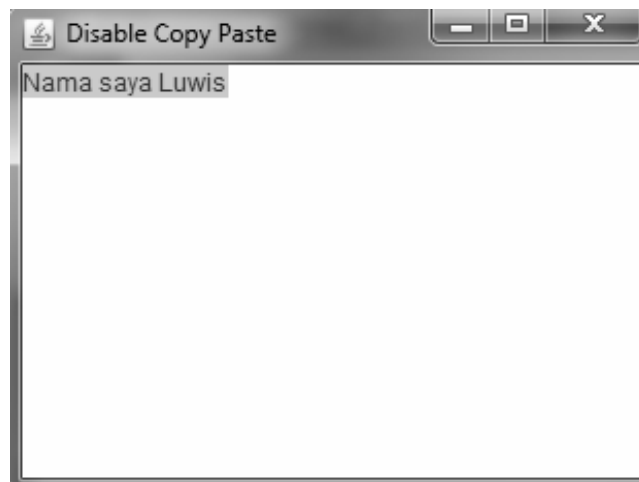
13. Buka class **Main.java**. Ketik kode berikut.

```
package disablecopypasteapp;  
  
public class Main {  
    public static void main(String[] args) {  
        new FormUtama();  
    }  
}
```

14. Jalankan project DisableCopyPasteApp.

15. Klik kanan project DisableCopyPasteApp > Run.

16. Coba lakukan copy dan paste di dalam TextArea.



Gambar 13.4 Aplikasi DisableCopyPasteApp